

## The Systems Life Cycle

- 7.1 Analysis
- 7.2 Design
- 7.3 Development and Testing
- 7.4 Implementation
- 7.5 Documentation
- 7.6 Evaluation

## The Systems Life Cycle

Systems Analysts will review an existing system which is currently not meeting expectations. The steps shown below will be followed to develop a new system which is fit for purpose.

**Analysis:** Collecting information about the present system and identifying problems.

**Design:** Designing a new system to correct the problems identified in the analysis.

**Development & Testing:** Developing and testing new system.

**Implementation:** Replacing the old system with the new system.

**Documentation:** Creating technical and user documentation for new system.

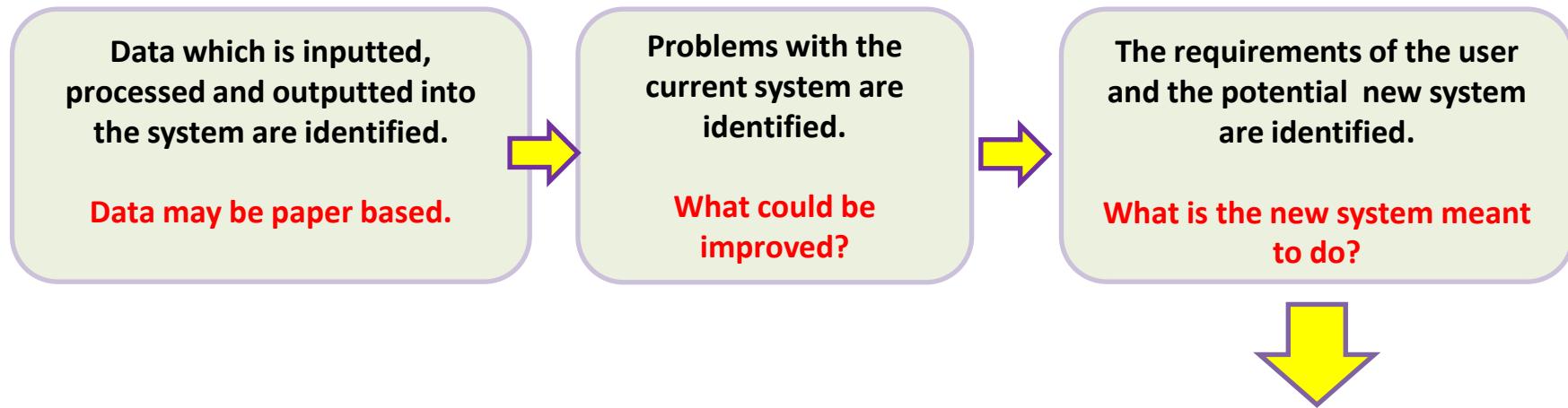
**Evaluation:** Evaluating whether the new system meets the requirements of the design requirements.

## The Systems Life Cycle

## 7.1 Analysis

Analysis techniques	Advantages	Disadvantages
<b>Observation</b> 	<p>The use of the current system is observed to find out how it works.</p>	<ul style="list-style-type: none"> <li>The analyst can obtain reliable information about the current system.</li> <li>Information will not be biased as it's the analyst's point of view.</li> </ul> <ul style="list-style-type: none"> <li>Some employees may feel uncomfortable being observed</li> <li>Employees may perform differently if they know they are being observed.</li> </ul>
<b>Interview</b> 	<p>People who use the system are interviewed to gauge their feedback.</p>	<ul style="list-style-type: none"> <li>Can ask open ended questions to gauge more feedback.</li> <li>Questions can be modified during interview.</li> </ul> <ul style="list-style-type: none"> <li>It can be a time consuming process.</li> <li>The interviewee can't remain anonymous with this method.</li> </ul>
<b>Questionnaire</b> 	<p>Questionnaires are distributed to employees/ customers to find out a range of opinions about the current system.</p>	<ul style="list-style-type: none"> <li>Questionnaires can reach a larger audience.</li> <li>Questions can be answered quickly using tick boxes – cheaper method of analysing.</li> </ul> <ul style="list-style-type: none"> <li>Questionnaires may not be completed accurately.</li> <li>May not be all returned.</li> <li>Can not expand or clarify answers to questions.</li> </ul>
<b>Collecting Documents</b> 	<p>Existing documentation for the current system is analysed to identify what data is inputted and outputted.</p>	<ul style="list-style-type: none"> <li>Analyst's can see how current the paper system operates.</li> <li>Can obtain information critical to the system.</li> </ul> <ul style="list-style-type: none"> <li>It can be a time consuming process.</li> <li>Due to the amount of time required this is a costly method.</li> </ul>

### 7.1 Analysis



#### New System Requirements Specification

- Once the systems analysts have completed the analysis stage of the systems life cycle they should be fully aware of the limitations of the current system.
- The next step will be to design a new system (**normally computer based**) to resolve the problems identified by the users and the systems analyst.
- The Requirements Specification will be created which will outline the required improvements and expectations for the new system.

### 7.2 Design

Based on the information from the analysis stage a new system will be designed. The new system will hope to resolve the issues identified in the current system.

Users should be able to interact with a system via a data capture form so is it easier and quicker to input data.

The screenshot shows a data capture form for a book record. The form includes fields for BookID (1001), Book Name (Catching Fire), Author\_ID (SC450), Genre (Adventure), Shop (ama222), Current Stock (5), Sale (checkbox checked), Sold (45), and Price (£4.00). To the right of the form is a thumbnail image of the book 'Catching Fire' by Suzanne Collins. Below the form are buttons for Add Record, Save Record, and Delete Record.

**Tick boxes/radio buttons to enter choices**

**Drop down lists to choose an option**

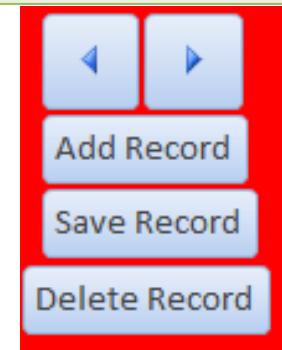
A screenshot of a drop-down list box showing genre options: Adventure, Adventure, Crime, Drama, Fantasy, and Science Fiction. The 'Adventure' option is listed twice, and 'Crime' is the currently selected item.

#### On-screen forms should:

- Appropriate spacing for each field.
- Screen filled/not too much white space.
- Clearly defined input area for each field.
- An easy to read font/font size.
- A sensible font colour/background colour.
- Drop down lists and tick boxes.
- No overlapping of items.
- Navigation aids

#### Buttons:

- Go forward or backwards
- Add, Save and delete records



Field Name	Data Type
BookID	Number
Book Name	Text
Author_ID	Text
Genre	Text
Shop	Text
Current Stock	Number
Sale	Yes/No
Sold	Number
Price	Number
Picture	OLE Object

General	Lookup
Field Size	Double
Format	Currency
Decimal Places	Auto



Before a data input form can be created the database has to be created. The following need to be considered:

- Type of data which will be captured and inputted.
- Different tables which may be required and relationships which could be established.
- Different fields (including Key) which would be required.
- Data types that would be required for each field
  - Text
  - Integer (numeric)
  - Double
  - Boolean (Yes/No)
  - OLE Object (Image)

### Paper Based Forms Must Include:

- Clear heading and instructions.
- Use of text boxes to collect information such as name, DOB etc.
- Use of tick boxes to make choices easier (Gender).
- Clear fonts and text styles.

### 7.2 Design

- **Data Validation** checks whether the **data entered** fulfils **specific criteria** to ensure the data is valid.
- Different **validation techniques** can be used on different fields depending on the **type of data** required for each field.



Presence Check	To check if the data has been entered ( <b>is it present</b> ).
Range Check	To check the data entered is in the correct range .
Length Check	To check if item of text is too short or too long ( <b>check if full number has been entered</b> ).
Type Check	To check if the type of data entered is correct ( <b>no letters in a numeric field</b> ).
Format Check	To check if the data has been entered in the correct format ( <b>Date – DD/MM/YYYY</b> )

**Data Verification:** Is a method of double checking the data to see if it is correct.

**Proof Reading:** Once data has been entered it will be checked to see if it is correct (e.g. check is name is spelt correctly).

**Double Entry:** The same data is required to be entered twice (e.g. type new password twice).



Change Password

Current Password*	<input type="text"/>
New Password*	<input type="text"/>
Confirm Password*	<input type="text"/>

### 7.3 Development and Testing

#### Development

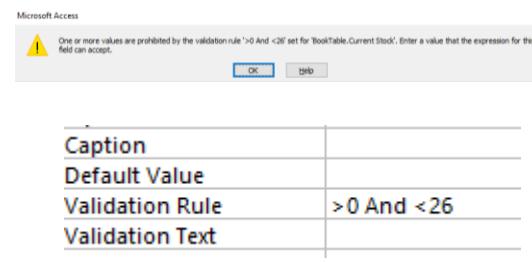
##### Create the file structure:

- Field Names
  - (To store specific data)
- Set Specific Data Types

Field Name	Data
BookID	Number
Book Name	Text
Author_ID	Text
Genre	Text
Shop	Text
Current Stock	Number
Sale	Yes/No
Sold	Number
Price	Number

##### Validation Techniques:

- Validation rules will ensure the correct data is entered into the fields.



##### User Interface:

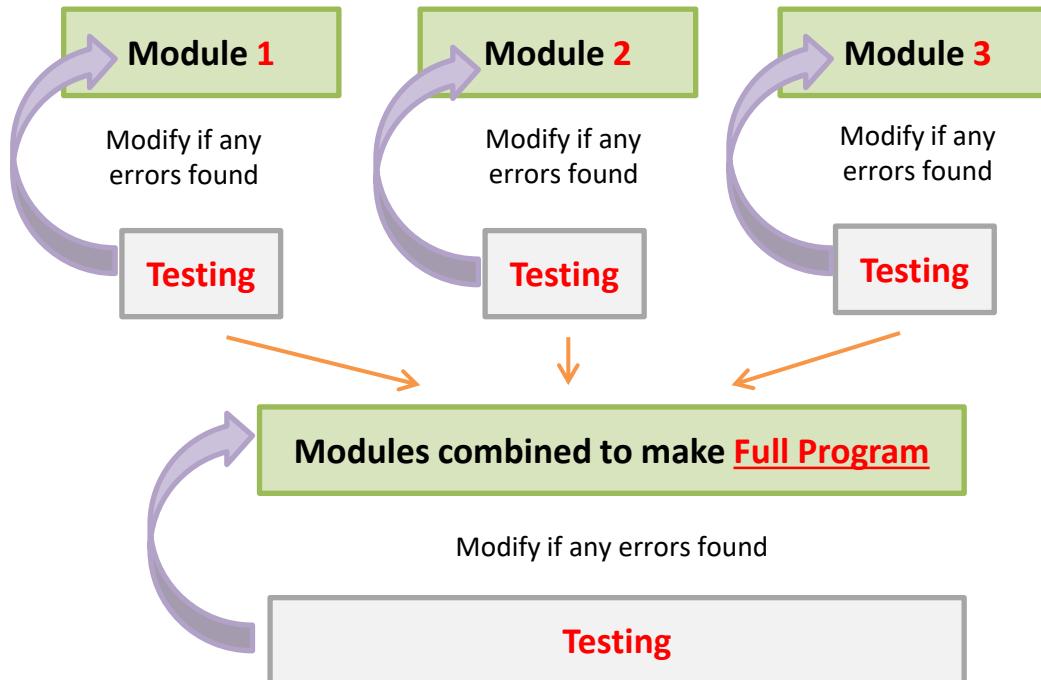
- The user interface will allow the user to interact with the system and input data.

Each step in the development stage will be checked to ensure there are no errors in the system.

- **Validation rules** will be checked to see if only the correct values can be entered into the relevant fields.
- **User Interface:** Check to see if they are user friendly and to see if the different components work (navigation buttons).

### 7.3 Development and Testing

#### Testing



#### Modifications which could be made:

- Data/file structures
- Validation Rules
- Input methods & output formats

Testing takes place to ensure all components of the system are working correctly before the system is implemented.

- Software systems are developed in **Modular form** which means each part of the system is **developed separately by the programmer**.
- This allows each module of the system to be **tested separately** once it has been developed.
- If **errors are found** in the **modules or full program** then then **modifications will be made to the system**.

## 7.3 Development and Testing

**A typical test plan would include:**

- What is being tested
- The test data that will be used
- The expected outcome of the Test.

Below are examples of **three different types of test data** that could be entered into the system. Once the system has been fully tested then **live data** (actual data that would be entered) will be entered into the system. **Results are compared to those produced from the current system.**

### Normal

Normal data – data within a (given) range

Example – 10% - 90%

### Abnormal

Abnormal data – data outside the **range** (limits)

Example – Less than 0% or more than 100%

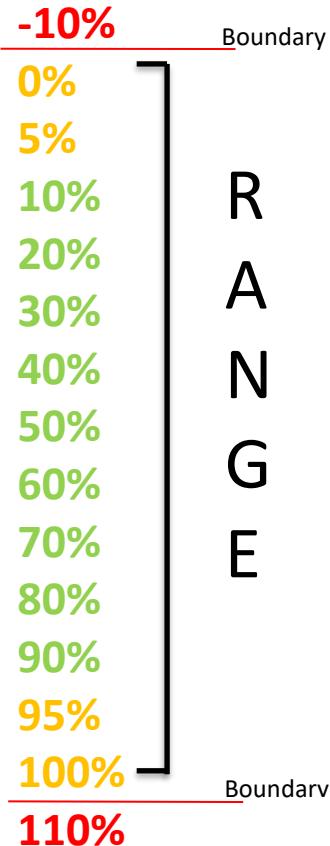
### Extreme

Extreme data – data on the boundaries of the range or at the limits of acceptability.

Example – 0%-5% or 95-100%

### Example: Test Scores

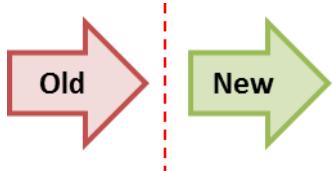
0 – 100% (Range)



## The Systems Life Cycle

### 7.4 Implementation

#### Direct Changeover



The existing system is **stopped** and **replaced by the new system immediately**.

The data that was inputted into existing system is now inputted into the new system.

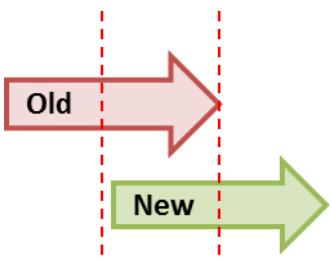
#### Advantages

- The whole system would have been fully tested before it is implemented so less chance of errors.
- Costs are reduced as only one system is being used. Tasks are not duplicated.

#### Disadvantages

- If the new system fails then old system is not available to fall back to.
- There may not be enough time to provide training to employees.

#### Parallel Running



The **existing and new system work together for a period of time** until the new system fully takes over.

Data is inputted into both systems whilst they are running at the same time.

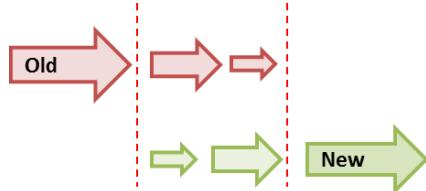
- If the new system does not meet the requirements then old system is still available.
- Employees can be trained gradually how to used the new system.

- Tasks will be duplicated as data is inputted into both systems.
- This will require more employees which will result in more costs (paying salaries) for the employers.

## The Systems Life Cycle

### 7.4 Implementation

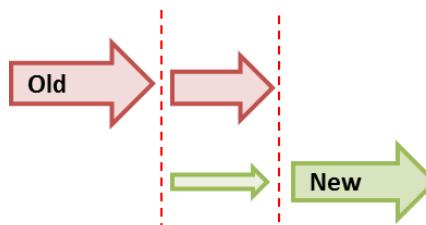
#### Phased Implementation



The new system is **gradually introduced**. When parts of the new system (Module) are working at a satisfactory level then more elements of the system are phased in.

Eventually the **old system will be phased out over a period of time**.

#### Pilot Running



The **new system is piloted (trialed) in one part (Branch)** of an organisation. If the pilot is successful then it will be implemented across all branches.

**Example:** If a new system is introduced in book store which is part of a chain. Then eventually if the system is successful it will be introduced to all of the books stores in that chain.

#### Advantages

- You only introduce a new part of system if the current phase is working correctly.
- If a part of the new system does not meet the requirements then you can always go back to point of system which was working.

#### Disadvantages

- Time consuming as every part of the system which is implemented needs to be evaluated.
- A number of training sessions would be required for employees when each step of the new system is introduced.

- If the new system does not meet the requirements then only one branch is affected.

- Training can take place in one branch at a time. Then employees could be used in other branch to help with training.

- It could take a while for the whole system to be implemented across all branches.
- If the system fails in one branch then modifications and further training would be required resulting in delays.

## 7.5 Documentation

There are two types of documentation that should be produced when creating a new system:

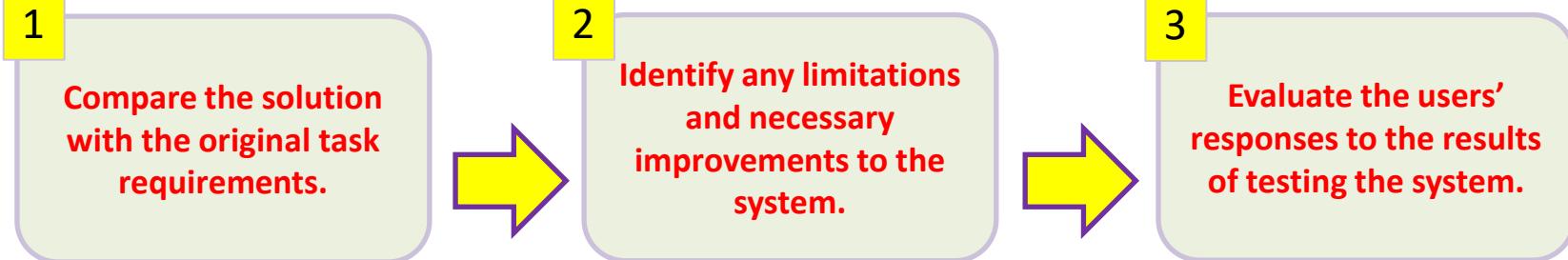
- **User Documentation**
- **Technical Documentation**

User Documentation	Technical Documentation
<p>The user documentation is intended to help the <b>users of the system</b>.</p> <p><b>User documentation usually includes:</b></p> <ul style="list-style-type: none"> <li>• List of minimum hardware and software required to use the system</li> <li>• How to install the system</li> <li>• How to start / stop the system</li> <li>• How to use the features of the system</li> <li>• Screenshots showing the system in typical use</li> <li>• Example inputs and outputs</li> <li>• Explanations of any error messages that might be shown</li> <li>• A troubleshooting guide</li> </ul>	<p>The maintainers are usually technical people, who need to know exactly how the system works.</p> <p><b>Technical documentation usually includes:</b></p> <ul style="list-style-type: none"> <li>• Details of the hardware and software required for the system</li> <li>• Details of data structures (data types, field names, etc.)</li> <li>• Details of expected inputs</li> <li>• Details of validation checks</li> <li>• Details of how data is processed</li> <li>• Diagrams showing how data moves through the system</li> <li>• Flowcharts describing how the system works</li> </ul>



### 7.6 Evaluation

The final stage of the system life cycle is to **evaluate** the new system which has been implemented in full. The purpose of the evaluation is refer back to the **requirements specification** to see whether the new system has **resolved the issues of the previous system** and **met the requirements** stated in the **design**.



- After the evaluation is complete if there are any problems or limitations with the system from a technical or user point of view then the system analyst will refer back to the design stage of the systems life cycle.
- Based on the feedback necessary adjustments could be made to improve the overall efficiency and usability of the new system